УДК 536.244:621.438

ШИФРУВАННЯ ЗІ ЗБЕРЕЖЕННЯМ ФОРМАТУ

М.А. Тузовська[1]

[1] *Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Фізико-технічний інститут*

Анотація

Аналіз перестановок на невеликій множині та шифруваня що зберігає формат, еквівалентність з максимально незбалансованою мережею Фейстеля.

Ключові слова: перемішування карток, симетричне шифрування, шифрування зі збереженням формату, перестановка Торпа, незбалансована мережа Фейстеля

Вступ

Досить часто виникає потреба у шифруванні за допомогою симетричного шифру зовсім невеликих повідомлень як то 5-значний поштовий індекс, 4-значний PIN-код або 10-значний ідентифікаційний код, але за умови, що шифротекст також буде мати невеликий розмір (наприклад, з міркувань оптимізації обсягу використовуваної пам'яті). Тобто, без втрати загальності необхідним є застосування симетричного шифру $E : K \times M \to M$, де $M = \{0, 1, \ldots, N-1\}$ для невеликого значення натурального числа N. Використання ендоморфного шифру також не обмежує загальності, тому вважатимемо, що для довільного значення ключа $k \in K$ відображення $E_k = E(k, \cdot)$ є перестановкою на множині $\{0, 1, \ldots, N-1\}$. Одним з можливих варіантів вирішення цієї задачі є використання відображення множини повідомлень M у множину повідомлень одного з відомих блокових шифрів, наприклад, шифру AES, використання самого шифру для шифрування, а потім використати обернене відображення результату шифрування у множину повідомлень M. Але через те, що значення N є достатньо невеликим, використання відомих шифрів, у яких потужність множини повідомлень є значно більшою, ставить під сумнів стійкість таких конструкцій.

Згадана вище проблема шифрування невеликого простору була досліджена Блеком та Рогвеєм [9], але автори не змогли знайти практичних доведень безпеки для N значень, де $q > \sqrt{N}$ запитів та шифрування потребує N обчислювальних кроків, тоді коли $2^{20} < N < 2^{50}$. У цій статті наведені розв'язки даної проблеми. Підхід базується на перестановці Торпа. Гарним розв'язком буде реалізувати випадкове перемішування.

1. Види перестановок на невеликій множині

Деякі способи перемішування карт призводять до схем шифрування. Для гарного перемішування карт потрібна ідея, яку запропонув Наор [10]: можна простежити траєкторію карти, не звертаючись до інших карт в колоді. Більшість звичайних перетасовок, таких як riffle, залежать від попередніх кроків, або від інших карт у колоді. Перестановка Торпа і Swap-or-not – не залежать.

Перестановка Торпа для N карт. Послідовність карт ділиться на дві рівні частини: по $N/2$ кожна. Далі, обирають дві перші карти з лівої та правої частини і виконують перестановку, в залежності від значення, отриманого в результаті підкидання чесної монети. Тобто, кожна пара карт на позиціях x та $x + N/2$, $x \in \{1, 2, ..., N/2\}$ будуть переставлятись відповідно значенню функції, яка генерує випадковий біт C, який визначить, переставляти пару місцями чи ні.

Для зручності обирають $N = 2^n$, тобто N є потужністю 2.

Алгоритм перестановки Торпа. Th[N, R], $N = 2^n$ Для перемішування колоди з N карт, для раундів r, $r \in \{1, 2, ..., R\}$:

1. Потрібно розділити множину N навпіл.

2. Використовуючи випадковий біт C, згенерований незалежно, такий що: $(P(C = 0) = 1/2 = P(C = 1))$, для карт на позиціях x та $x + N/2$ випадковий біт C визначає, карти переставляємо на місця $2x$ та $2x + 1$, якщо $(C = 0)$, або $2x$ та $2x + 1$, якщо $(C = 1)$.

Інверсія Th[R,N] буде мати вигляд: $\{Z(l, t) : l \in \{0, 1\}^{d-1}, t \in \{0, 1, ...\}\}$ – послідовність випадкових бернулевих величин $\sim B(1/2)$.

Внесок алгоритму випадкового пермішування Торпа для криптографії та теорії складності. Наор [10] помітив, що перемішування Торпа не залежить від карт у колоді та попередніх кроків. Тобто, можна відтворити маршрут до будь-якої карти, не звертаючи увагу на усі інші карти в колоді. Якщо перемішування Торпа виконує випадкову перестановку достатньо швидко, то ця властивість робить його актуальним для шифрування у невеликому просторі. Випадковий біт C використовується для карт x та $x + N/2$ у раунді r, C визначений випадковою функцією F, заданою певним ключем K для x та r. Рядок K

компактно викликає усі $N/2 \cdot R$ випадкових бітів – достатньо, щоб перемішати усю колоду випадковим чином.

Шифрування n-бітового рядка Swap-or-not. Визначимо N точок – це розміщення карт на позиціях $X \in N$, де $N\{0, 1, ..., N-1\}$. Після перемішування колоди переглядають положення карти на позиції X. Це перетворення і є виходом Y від X. Секретним ключем має бути випадковість, що виникла у момент перемішування.

Простір повідомлень - це набір $X = \{0, 1\}^n$, r — раундовий ключ блочного шифрування KF та послідовність $K_1, ..., K_r \in \{0, 1\}^n$ підключів для раундових функцій $F_1, ..., F_r$, кожна з яких відображає n-бітовий рядок в 1 біт: F_i: $\{0,1\}^n \to \{0, 1\}$.

Алгоритм шифрування Swap-or-not.

1. for $i \leftarrow$ to r do
2. $X' \leftarrow K_i \oplus X$, $X \leftarrow \max(X,X')$
3. if $F_i(\hat{X}) = 1$ then $X \leftarrow X'$
4. return X

Отримуємо перестановку $K \to K_i \oplus X$, {X, $K_i \oplus X$}. Розшифрування Swap-or-not ідентичне шифруванню: виконуємо алгоритм від r-кроку до першого кроку.

Кожен відкритий текст відображається в шифрований, шляхом операції XOR з ключами $K_1, ..., K_r$.

Припустимо, що у нас є N карт, по одній на кожну позицію. $X \in \mathbb{N}$, де $N = 2^n$. Для перемішування обираємо випадковий $K \in \{0, 1\}^n$, а потім кожну пару позицій карток X та $K \oplus X$, повертають, в залежності від значення результату підкидання чесної монети. Якщо значення монети — 1, переставляємо, інакше — залишаємо на вхідних позиціях. Процес можна повторити будь-яким числом разів r, використовуючи незалежні значення підкинутої монети для кожного перемішування:

1. $K \leftarrow \{0, 1\}^n$
2. for each pair of positions $\{X, K \oplus X\}$
3. $b \leftarrow \{0, 1\}$
4. if $b = 1$ then swap the cards at the positions X and $K \oplus X$

Перемішування колоди $N = 2^n$ карток, кожна з яких на позиції $X \in \{0, 1\}^n$. В алгоритмі показано один раунд перестановки. Для гарного перемішування, слід використати більшу кількість раундів.

2. Практична реалізація з FPE

Існує багато альтернативних поглядів на те, що відбувається в алгоритмі перемішування Торпа, та найбільш вагомим для криптографів є те, що алгоритм поводить себе як максимально незбалансована мережа Фейстеля.

Th[N, R] = максимально незбалансована мережа Фейстеля.

Розглянемо мережу Фейстеля для $N = 2^n$. Можливі атаки:

1. для визначеної кількості раундів — $2^{n/2}$
2. для r раундів — $2^{n/2+\log_{10} R}$

Для Th[N, R], $N = 2^n$, для раунду r переміщеня картки в положення $x \in \{0, ..., N-1\}$ до позиції:

$$\begin{cases} 2x + F_k(r, x), \text{якщо } x \le N/2 \\ 2(x - N/2) + (1 - F_k(r, x - N/2)), \text{інакше} \end{cases}$$

— еквівалентно мережі Фейстеля (Рисунок 1)

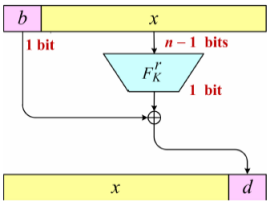

Рис. 1. Один раунд максимально незбалансованої мережі Фейстеля

У незбалансованій мережі Фейстеля [11, 5], ліва і права частини в n-бітовому рядку, в якому виконуються дії – можуть мати різні довжини. Для максимально незбалансованої мережі Фейстеля має-ться на увазі, що функція округлення забирає $n-1$ біт і повертає один біт, який визначатиме перетворення. Якщо ця функція визначить випадкові біти, незалежні один від одного, то для кожного раунду перестановки – незбалансована мережа Фейстеля і буде перемішуванням Торпа.

3. Практична реалізація алгоритму випадкового перемішування Торпа.

Більш ефективною реалізацією Th[N, R] є техніка, яка дозволить одним викликом забезпечити п'ять раундів шифрування або дешифрування (Рисунок 2).

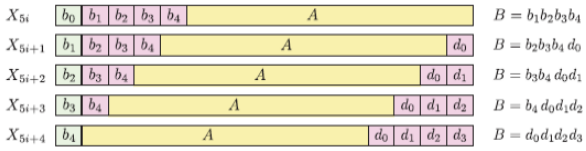

Рис. 2. Приклад реалізації п'ятикратного прискорення детермінованого шифрування алгоритмом Торпа.

Виклики AES були б необхідні, щоб зробити різну кількість проходів над рядками різних розмірів.

Послідовні n-бітові рядки x_j (Рисунок 2), шифруємо піднімоючись вгору або розшифровуємо спускаючись вниз. Для будь якого $A \in \{0, 1\}^{n-5}$ та для раунду j яке ділиться на 5, один виклик обчислює C для усіх $(n-1)$ – бітових рядків:

A для раунду j: * * * * A
A для раунду j + 1: * * * A *
A для раунду j + 2: * * A * *
A для раунду j + 3: * A * * *
A для раунду j + 4: A * * * *
(*) - може бути 0 або 1

X записується $X[i, ..., j]$, $i > j$. Якщо $v_1, ..., v_k$ бітовий рядок або ціле число, $v_1, ..., v_k$ – це кортеж $(v_1, ..., v_k)$, закодований деяким фіксованим способом.

Позначимо зашифрований текст $X \in \{0, 1\}^n$ після того як i раундів Th[N, R] з X_i, $X_0 = X$. Замість того щоб оцінювати p в $X_i[1, \ldots, n-1]$, $i > j$, – витягнемо достатню кількість бітів, щоб визначити усі $C(U, r)$, які можуть бути необхідні в тій самій групі із п'яти послідовних раундів. Реалізація цієї ідеї дещо складна, оскільки важливо щоб кожна монета $C(U, r)$ взята з виходу р була чітко визначена, як це значення може виникнути і водночас бути незалежним з $C(V, s)$, $C(V, s) = C(U, s)$

Стратегія проілюстрована на рис. 2. Використаємо той факт, що для $j \in \{0, 1, 2, 3, 4\}$, рядки X_{5i}, X_{5i+j} мають $(n-5)$-бітовий спільний рядок з A.

Реальна складність виникає тоді, коли N є потужністю 2. Ретельно узагальнюючи, ідея виконується шляхом заміни рядкових операцій арифметикою за модулем, дає початок п'ятикратному прискоренню для Th[N, R] будь-якого числа точок N за умови, що N кратна 32. П'ятикратне прискорення використовує $5 \cdot 2^4 = 80$ з 128 бітів виходу PRF. Це узагальнення того, що k-кратне прискорення виводиться, якщо PRF-вихід рівний $k \cdot 2^{k-1}$ бітів, хоча це вимагає округлення N до наступного кратного від $2 \cdot k$.

Висновки

Розглянено термін FPE, та проблематику, чому ми використовуємо шифрування зі збереженням формату. Багато систем використовують FPE, тому що даний вид шифрування — гарне наближення для сім'ї рівномірних перестановок має гарні статистичні властивості. У даній роботі визначені алгоритми перемішування, шифрування.

Алгоритм Торпа набуває найбільшої ефективності при модифікації п'ятикратного прискорення [3].

Перелік використаних джерел

1. Dowkin M.: Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption. NIST Special Publication 800-38G, Gaithersburg (2016)
2. Bellare, M., Ristenpart, T., Rogaway, P., Stegers, T.: Format-Preserving Encryption. In: Jacobson, M.J., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 295–312. Springer, Heidelberg (2009)
3. Hoang V., Morris B., Rogaway P. (2018) An Enciphering Scheme Based on a Card Shuffle. In: Annual Cryptology Conference CRYPTO 2012: Advances in Cryptology – CRYPTO 2012.
4. Bellare, M., Rogaway, P., Spies, T.: The FFX mode of operation for format-preserving encryption (2010). In: submission to NIST.
5. Brightwell, M., Smith, H.: Using datatype-preserving encryption to enhance data warehouse security. In: 20th National Information Systems Security Conference Proceedings (NISSC), pp. 141–149 (1997)
6. Granboulan, L., Pornin, T.: Perfect Block Ciphers with Small Blocks. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 452–465. Springer, Heidelberg (2007)
7. Hoang, V.T., Rogaway, P.: On Generalized Feistel Networks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 613–630. Springer, Heidelberg (2010)
8. Luby, M., Rackoff C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM J. on Computing 17(2), 373–386 (1988)
9. J. Black and P. Rogaway. Ciphers with arbitrary finite domains. Topics in Cryptology – CT-RSA 2002, LNCS vol. 2271, Springer, pp. 114–130, 2002.
10. M. Naor and O. Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited. J. of Cryptology, 12(1), pp. 29-66, 1999.
11. S. Lucks. Faster Luby-Rackoff ciphers. Fast Software Encryption (FSE 1996), LNCS vol. 1039, Springer, pp. 180–203, 1996.

‘